CLAIMS

What is claimed is:

1. A method for secure communications in a computer network, comprising;

combining individually encrypted network security protection handshake messages into a set of encrypted messages wherein each encrypted handshake message is derived using a public key containing an encryption exponent;

determining a root node of a binary tree comprising leaf nodes corresponding to each encryption exponent;

calculating a product of the encrypted messages;

extracting at least one root from the product of the encrypted messages; and

decrypting the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes decreasing the number of modular inversions wherein efficiency of the decryption is increased.

2. The method of claim 1, wherein the secure communications include secure socket layer ("SSL") messages.

3. The method of claim 1, wherein the secure communications include transport layer security ("TLS") messages.

4. The method of claim 1, wherein the secure communications include internet protocol secure ("IPSec") techniques.

5. The method of claim 1, wherein evaluating the at least one promise includes multiplying an inversion of a total product of the leaf nodes with a partial product of the leaf nodes to produce the inversion of an individual leaf node.

6. The method of claim 1, further comprising minimizing the disparity among the sizes of the encryption exponents of the public keys within the set.

7. The method of claim 1, wherein determining includes using a plurality of separate, parallel batch trees finding the root node of each tree and combining the final answers.

8. The method of claim 1, wherein decrypting includes simultaneous multiple exponentiation such that the encryption exponents are combined to reduce the number of exponentiations.

9. A method for improving secure communications in a computer network comprising;

combining individually encrypted network security protection handshake messages into a set of encrypted messages wherein each encrypted handshake message is derived using a public key containing an encryption exponent;

determining a root node of a binary tree comprising leaf nodes corresponding to the encryption exponent of each encrypted message;

calculating a product of the encrypted messages;

extracting at least one root from the product of the encrypted messages; and

decrypting the encrypted messages by evaluating at least one individual leaf node by multiplying an inversion of the total product of leaf nodes with a partial product of the leaf nodes to produce an inversion of the at least one individual leaf node wherein efficiency of the decryption is increased.

10. The method of claim 9, wherein the network security protection handshake messages include secure socket layer ("SSL") messages.

11. The method of claim 9, wherein the network security protection messages include transport layer security ("TLS") messages.

12. The method of claim 9, wherein the network security protection messages include internet protocol secure ("IPSec") messages.

13. The method of claim 9, further comprising minimizing the disparity among the sizes of the encryption exponents of the public keys within the set.

14. The method of claim 9, wherein determining includes using a plurality of separate, parallel batch trees finding the root node of each tree and combining the answers.

15. The method of claim 9, wherein decrypting includes simultaneous multiple exponentiation such that the encryption exponents are combined to reduce the number of exponentiations.

16. The method of claim 9, wherein decrypting includes expressing the at least one root as at least one promise and evaluation the at least one promise at the leaf nodes decreasing the number of modular inversions.

17. A method for secure communications in a computer network, comprising;

combining individually encrypted network security protection handshake messages into a set of encrypted messages wherein each encrypted handshake message is derived using a public key containing an encryption exponent;

determining a root node of a binary tree comprising leaf nodes corresponding to the encryption exponent of each encrypted message;

calculating a product of the encrypted messages;

extracting at least one root from the product of the encrypted messages; and

decrypting the encrypted messages by minimizing the disparity between the sizes of the encryption exponents of the public keys, wherein efficiency of the secure communications is increased.

18. The method of claim 17, wherein combining includes secure socket layer ("SSL") messages.

19. The method of claim 17, wherein combining includes transport layer security ("TLS") messages.

20. The method of claim 17, wherein combining includes internet protocol secure ("IPSec") messages.

21. The method of claim 17, wherein determining uses a plurality of separate, parallel batch trees finding the root node of each tree and combining the final answers.

22. The method of claim 17, wherein decrypting includes simultaneous multiple exponentiation such that the encryption exponents are combined to reduce the number of exponentiations.

23. The method of claim 17, wherein decrypting includes expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes decreasing the number of modular inversion.

24. The method of claim 17, wherein decrypting includes evaluating at least one individual leaf node by multiplying an inversion of the total product of leaf nodes with a partial product of the leaf nodes to produce an inversion of the at least one individual leaf node.

25. A method for improving secure communications in a computer network, comprising;

combining individually encrypted network security protection handshake into a set of encrypted messages wherein each encrypted handshake message is derived using a public key containing an encryption exponent;

determining a root node of a binary tree comprising leaf nodes corresponding to each encryption exponent by using a plurality of separate parallel batch trees finding the root node of each tree and combining the final answers;

calculating a product of the encrypted messages;

extracting at least one root from the product of the encrypted messages; and

decrypting the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes

producing a reduced number of modular inversions wherein efficiency of establishing secure communications is increased.

26. The method of claim 25, wherein combining includes secure socket layer ("SSL") messages.

27. The method of claim 25, wherein combining includes transport layer security ("TLS") messages.

28. The method of claim 25, wherein combining includes internet protocol secure ("IPSec") messages.

29. The method of claim 25, wherein decrypting includes simultaneous multiple exponentiation such that the encryption exponents are combined to reduce the number of exponentiations.

30. The method of claim 25, wherein evaluating the at least one promise includes multiplying an inversion of a total product of the leaf nodes with a partial product of the leaf nodes to produce the inversion of an individual leaf node.

31. The method of claim 25, further comprising minimizing the disparity among the sizes of the encryption exponents of the public keys within the set.

32. A method for secure communications in a computer network, comprising;

combining individually encrypted network security protection messages into a set of encrypted messages, wherein each encrypted handshake message is derived using a public key containing an encryption exponent;

determining a root node of a binary tree comprising leaf nodes corresponding to each encrypted messages encryption exponent;

calculating a product of the encrypted messages;

minimizing the disparity among the sizes of the encryption exponents of the public keys within the set;

extracting at least one root from the product of the encrypted messages; and

decrypting the encrypted messages by evaluating the at least one leaf node by multiplying an inversion of a total product of the leaf nodes with a partial product of the leaf nodes to produce the inversion of the at least one leaf node wherein efficiency of establishing secure network communications is increased.

33. The method of claim 32, wherein combining includes secure socket layer ("SSL") messages.

34. The method of claim 32, wherein combining includes transport layer security ("TLS") messages.

35. The method of claim 32, wherein combining includes internet protocol secure ("IPSec") messages.

36. A method for secure communications in a computer network, comprising:

coupling a client to a web server;

sending a client hello message to the web server;

generating a public / private key pair at the web server, wherein the public key contains an encryption exponent;

responding to the client with a server hello message comprising the public key;

encrypting a random handshake message at the client using the public key;

sending the encrypted handshake message to a batch-decryption server;

batching handshake messages on a batch-decryption server according to the public key such that the disparity between the sizes of the encryption exponents of the public key is minimized;

separating the batch's e^{th} root in a downward-percolation phase into constituent decrypted messages, wherein internal inversions are converted to modular divisions increasing efficiency by producing a reduced number of modular inversions;

scheduling the batch-decryption server based on server-load considerations;

decrypting the handshake messages using at least one alternate

expression of at least on arithmetic function of at least one batch's e^{th} root; and sending the decrypted message to the web server.

37. The method of claim 36, wherein batching handshake messages includes Secure Socket Layer ("SSL") messages.

38. The method of claim 36, wherein combining includes transport layer security ("TLS") messages.

39. The method of claim 36, wherein combining includes internet protocol secure ("IPSec") messages.

40. The method of claim 36, wherein batching further comprises an upward-percolation phase that combines individual encrypted messages to form a value, v wherein v is the product of the individual encrypted messages raised to the power of e/e_1, e being the product of all individual encryption exponents e_1.

41. The method of claim 36, wherein the value v is determined by the equation $v = \prod_{i=1}^{b} v_i^{e/e_i}$, where e is the product of individual exponentiation exponents, v_i is the individual encrypted message, e_i is the individual public key, and b is the number of encrypted messages in a particular batch.

42. The method of claim 36, wherein batching further comprises an exponentiation phase that includes the extraction of an e^{th} root from the value, v.

43. The method of claim 36, wherein exponentiation further includes simultaneous multiple exponentiation such that the encryption exponents are combined to reduce the number of exponentiations.

44. The method of claim 36, wherein exponentiation includes combining a plurality of inversions to form a single modular inversion.

45. The method of claim 36, wherein decrypting includes reducing each encrypted batch message into a separate moduli, using separate parallel batch trees to determine the moduli, and combining the final answers.

46. A method for batch decryption in a computer network comprising:

combining a plurality of encrypted messages into a plurality of batches, wherein each encrypted message includes a public / private key pair, each public key comprising an encryption exponent;

scheduling the batches of encrypted messages using a plurality of criteria selected from a group including maximum throughput, minimum turnaround-time, minimum turnaround-time variance, and server load considerations, wherein the efficiency of establishing secure communications is

enhanced; and

replacing at least one inversion of at least one batch decryption operation with a single inversion and a plurality of multiplication operations, wherein the speed of the decryption is significantly improved.

47. The method of claim 46, wherein combining a plurality of encrypted messages includes secure socket layer ("SSL") messages.

48. The method of claim 46, wherein combining a plurality of encrypted messages includes transport layer security ("TLS") messages.

49. The method of claim 46, wherein combining includes internet protocol secure ("IPSec") messages.

50. The method of claim 46, further comprising using separate, parallel batch trees and combining the results.

51. The method of claim 46, wherein combining includes selecting the encrypted messages for the batches by balancing the encryption exponent.

52. A method for secure communications in a computer network, comprising;

combining individually encrypted network security protection handshake messages into a set of encrypted handshake messages wherein each

encrypted message is derived using a public key comprising an encryption exponent;

determining a root node of a binary tree containing leaf nodes corresponding to each encrypted message encryption exponent by using a plurality of separate parallel batch trees finding the root node of each tree and combining the final answers;

minimizing the disparity between the sizes of the encryption exponents of the public keys within the set;

using simultaneous multiple exponentiation such that the encryption exponents are combined to reduce the number of exponentiations;

calculating a product of the encrypted messages;

extracting at least one root from the product of the encrypted messages; and

decrypting the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes, and multiplying an inversion of a total product of the leaf nodes with a partial product of the leaf nodes decreasing the number of modular inversions by producing an inversion of the leaf node wherein efficiency of secure communications is increased.

53. The method of claim 52, wherein combining encrypted network security protection handshake messages includes secure socket layer ("SSL") messages.

54. The method of claim 52, wherein combining encrypted network security protection handshake messages includes transport layer security ("TLS") messages.

55. The method of claim 52, wherein combining encrypted network security protection handshake messages includes internet protocol secure ("IPSec") messages.

56. A method for performing batch decryption in a computer network, comprising:

receiving a plurality of encrypted messages generated using a plurality of public keys, wherein the plurality of public keys share a common modulus;

forming a binary tree using leaf nodes corresponding to the plurality of public keys;

placing each of the plurality of encrypted messages in a leaf node having a corresponding public key;

percolating the plurality of encrypted messages up the binary tree to form a root node including a product of the encrypted messages, extracting at least one root from the product of the encrypted messages by forming an exponentiation product in the root node;

expressing the at least one root using at least one promise that includes at least one alternative representation of at least one arithmetic function of the at least one root;

percolating the at least one root down the binary tree using the at least

one promise; and

decrypting the plurality of encrypted messages by evaluating the at least one promise at the leaf nodes, wherein efficiency of the decryption is increased by reducing a number of modular inversions and a number of root extractions.

57. The method of claim 56, wherein receiving a plurality of encrypted messages includes secure socket layer ("SSL") messages.

58. The method of claim 56, wherein receiving a plurality of encrypted messages includes transport layer security ("TLS") messages.

59. The method of claim 56, wherein receiving a plurality of encrypted messages includes internet protocol secure ("IPSec") messages.

60. The method of claim 56, wherein evaluating the at least one promise uses batched division to calculate a plurality of inverses for the plurality of leaf nodes using a single modular inversion, wherein the single modular inversion is multiplied with a partial product at each leaf node to produce a corresponding inverse for the leaf node

61. The method of claim 56, further comprising:

reducing each of the plurality of encrypted messages modulo p and q;

generating two parallel batch trees modulo p and q; and

batching in each of the two parallel batch trees modulo p and q.

62. The method of claim 56 , wherein the percolating includes balanced exponents.

63. The method of claim 56, wherein the percolating includes simultaneous multiple exponentiation.

64. A method for secure communications in a computer network, comprising:

generating a Rivest-Shamir-Adleman ("RSA") public / private key pair at a web server;

coupling a client to the web server;

sending a client hello message to the web server requesting the establishment of a Secure Socket Layer ("SSL");

responding to the client with a server hello message containing the RSA public key;

encrypting a random string R, the pre-master secret at the client, using the RSA public key, wherein the resulting cipher-text, C, contains R;

sending the encrypted cipher-text message, C, to the web server;

combining individually encrypted secure socket layer ("SSL") encrypted cipher-text messages to form a batch;

decrypting the batch of cipher-text, C, messages at the web server using the RSA private keys to determine R, wherein the efficiency of the decryption is enhanced by replacing at least one inversion with at least one multiplication; and

establishing a common session key between the web server and the client using R.

65. The method of claim 64, wherein decrypting includes using at least one alternative representation of at least one arithmetic function to reduce to the number of inversions.

66. A system for secure communications in a computer network comprising:

at least one client processor;

at least one web server; and

at least one batch server coupled among the at least one client processor and the at least one web server, wherein the at least one batch server receives requests for decryption of a plurality of individually encrypted network secure protection handshake messages, aggregates the plurality of individually encrypted handshake messages into at least one batch wherein each encrypted message is derived by using an encryption exponent from an Rivest-Shamir-Adleman ("RSA") public / private key pair, forms a binary tree containing leaf nodes corresponding to each encryption exponent, extracts at least one root from a product of the encrypted messages, decrypts the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes, and multiplies an inversion of a total product of the leaf nodes with a partial product of the leaf nodes producing an

inversion of the leaf node decreasing the number of modular inversions, and responds to the requests for decryption with corresponding plain-text.

67. The system of claim 66, wherein the individually encrypted network secure protection handshake messages includes secure socket layer ("SSL") messages.

68. The system of claim 66, wherein the individually encrypted network secure protection handshake messages includes transport layer security ("TLS") messages.

69. The method of claim 66, wherein the individually encrypted network secure protection handshake messages includes internet protocol secure ("IPSec") messages.

70. The system of claim 66, wherein the batch server aggregates the plurality of encrypted messages base on criteria including maximum throughput, minimum turnaround time, and minimum turnaround time variance.

71. A system for secure communications in a computer network, comprising at least one client processor coupled among at least one web server, wherein the web server receives requests for decryption of a plurality of individually encrypted network security protection handshake messages, aggregates the plurality of individually encrypted handshake messages into at

least one batch wherein each encrypted message is derived using an encryption exponent from an Rivest-Shamir-Adleman ("RSA") public / private key pair, forms a binary tree containing leaf nodes corresponding to each encryption exponent, extracts at least one root from a product of the encrypted messages, decrypts the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes, and multiplies an inversion of a total product of the leaf nodes with a partial product of the leaf nodes producing an inversion of the leaf node decreasing the number of modular inversions, wherein efficiency of secure communications is increased.

72. A system of scheduling batch decryption in a computer network, comprising:

a plurality of client processors;

at least one web server;

at least one batch server coupled among the at least one web server and the plurality of client processors using a Rivest-Shamir-Adleman ("RSA") decryption algorithm, wherein the at least one batch server links the plurality of client processors to the at least one web server; and

a scheduler, wherein during a timed period the scheduler places arriving encrypted messages in a queue forming a batch, wherein the encrypted messages in the queue are decrypted upon completion of the timed period.

73. A system for secure network communications in a computer network, comprising at least one batch server coupled among at least one client processor and at least one web server, wherein the at least one batch server uses a Rivest-Shamir-Adleman ("RSA") batch algorithm to decrypt an aggregation of encrypted messages transferred among the at least one client processor and the at least one web server.

74. A system for secure computer network communications, comprising at least one client processor and at least one server processor wherein the server processor combines decryption requests of Secure Socket Layer ("SSL") messages into at least one batch and decrypts the at least one batch using a Rivest-Shamir-Adleman ("RSA") batch decryption algorithm.

75. A computer-readable medium, comprising executable instructions for establishing secure communications in a computer network which, when executed in a processing system, causes the system to:

combine individually encrypted network security protection handshake messages into a set of encrypted messages wherein each encrypted handshake message is derived using a public key comprising an encryption exponent;

determine a root node of a binary tree containing leaf nodes corresponding to each encrypted messages encryption exponent by using a plurality of separate parallel batch trees to find the root node of each tree and combine the final answers;

minimize the disparity between the sizes of the encryption exponents of

the public keys within the set;

combine the encryption exponents using simultaneous multiple exponentiation such that the number of exponentiations is reduced;

calculate a product of the encrypted messages;

extract at least one root from the product of the encrypted messages; and

decrypt the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes, multiplying an inversion of a total product of the leaf nodes with a partial product of the leaf nodes producing an inversion of the leaf node and decreasing the number of modular inversions, wherein efficiency of establishing secure communications is increased.

76. An electromagnetic medium, comprising executable instructions for establishing secure communications in a computer network which, when executed in a processing system, causes the system to;

combine individually encrypted secure network handshake messages into a set of encrypted handshake messages wherein each encrypted handshake message is derived using a public key comprising an encryption exponent;

determine a root node of a binary tree containing leaf nodes corresponding to each encrypted messages encryption exponent by using a plurality of separate parallel batch trees to find the root node of each tree and combine the final answers;

minimize the disparity between the sizes of the encryption exponents of the public keys within the set;

combine the encryption exponents using simultaneous multiple exponentiation such that the number of exponentiations is reduced;

calculate a product of the encrypted messages;

extract at least one root from the product of the encrypted messages; and

decrypt the encrypted messages by expressing the at least one root as at least one promise and evaluating the at least one promise at the leaf nodes, multiplying an inversion of a total product of the leaf nodes with a partial product of the leaf nodes producing an inversion of the leaf node, and decreasing the number of modular inversions wherein efficiency of establishing secure communications is increased.